IN THE CLAIMS:

Please cancel claims 2, 7, 8, 11, 12, 24, 25 and amend the claims as follows:

1. (Currently Amended): A method of ~~editing~~ processing inbound and outbound frames using an offload unit and a delegated connection table, comprising:

initializing an entry in the delegated connection table with connection state data including a connection table index corresponding to a connection selected by a TCP stack executed by a host CPU for processing by the offload unit;

receiving [[a]] the delegated connection table index at the offload engine, which is coupled to the host CPU;

receiving a prototype header and data for transmission from [[a]] the TCP stack;

accessing [[a]] the delegated connection table entry using the delegated connection table index;

computing a TCP checksum based on a portion of the data for transmission; and

outputting a frame ~~including~~ that includes the TCP checksum[[,]] and the portion of the data for transmission.

2. (Cancelled)

3. (Original): The method of claim 1, wherein the frame includes a received data acknowledgement number.

4. (Original): The method of claim 3, wherein the received data acknowledgement number is obtained from the delegated connection table entry.

5. (Original): The method of claim 3, wherein the received data acknowledgement number is updated when a data is received from the destination.

6. (Currently Amended): The method of claim 1, wherein the frame includes a TCP timestamp, and the TCP timestamp is stored in the delegated connection table entry and is updated when data is received from a destination connection.

7-8 (Cancelled):

9. (Original): The method of claim 1, further comprising computing an IPv4 header checksum when a delegated connection is an IPv4-based connection.

10. (Original): The method of claim 1, further comprising:

accessing the connection table entry;

computing a TCP checksum based on another portion of the data for transmission; and

outputting an additional frame including the TCP checksum and the other portion of the data for transmission.

11-12 (Cancelled)

13. (Currently Amended): A method of processing inbound and outbound frames using an offload unit and a delegated connection table and producing ~~receive~~ received data acknowledgements for output to a destination ~~using an offload unit~~, comprising:

initializing an entry in the delegated connection table with connection stat data including a connection table index corresponding to a connection selected by a TCP stack executed by a host CPU for processing by the offload unit;

receiving [[a]] an inbound TCP frame from a destination connection;

determining that the destination connection is a connection delegated for processing by the offload unit;

determining whether a sequence number in the TCP frame is consecutive relative to a sequence number stored in a delegated connection table; ~~and~~

processing the inbound frame at the offload unit if the sequence number is consecutive and uploading the inbound frame to the host CPU for processing if the sequence number is not consecutive; and

updating the sequence number stored in the delegated connection table.

14. (Currently Amended): The method of claim 13, further comprising:

determining that the received sequence number is greater than a threshold; and

transmitting a receive data acknowledgement to the destination.

15. (Currently Amended): The method of claim 13, further comprising:

determining that a timer has expired; and

transmitting a receive data acknowledgement to the destination.

16. (Currently Amended): The method of claim 13, further comprising:

determining that a count of unacknowledged received frames is greater than a limit; and

transmitting a receive data acknowledgement to the destination.

17. (Currently Amended): A method of processing inbound and outbound frames using an offload unit and a delegated connection table by ~~communicating receive data acknowledgement state from~~ [[an]] initializing an entry in the delegated connection table with connection state data including a connection table index corresponding to a connection selected by a TCP stack executed by a host CPU for processing by the offload unit [[to]] and communicating receive data acknowledgement state from an application program running on the host CPU, comprising:

updating connection state data stored in a delegated connection table; and

comparing a portion of the connection state data to a threshold to set a notification flag.

18. (Original): The method of claim 17, further comprising:

outputting a notification to the application program responsive to the notification flag value; and

updating at least a portion of the connection state data.

19. (Currently Amended): The method of claim 17, wherein the threshold is at least one of a timer value, a count of unacknowledged received frames or a received sequence number.

20 - 21. (Cancelled)

22. (Currently Amended): An apparatus for processing inbound and outbound frames using an offload unit and a delegated connection table and editing outbound frames, comprising:

means for initializing an entry in the delegated connection table with connection state data including a connection table index corresponding to a connection selected by a TCP stack executed by a host CPU for processing by the offload unit;

means for determining an IPv4 checksum;

means for determining a TCP checksum;

means for obtaining the connection state data for ~~a delegated~~ the selected connection; and

means for constructing a frame for transmission at least partially responsive to the ~~current~~ connection state data.

23. (Currently Amended): The apparatus of claim 22, wherein the state connection data includes a received sequence number, a TCP timestamp and a received data acknowledgement.

24 -25 (Cancelled)

26. (New): The method of claim 1 wherein connections that are not delegated to the offload unit or that require special processing are processed by the TCP stack; and

the offload unit may request legacy processing of one of the delegated connections by the TCP stack so that outgoing frames are transmitted by either the TCP stack or the offload connection.

27. (New): The method of claim 26 wherein a system memory is coupled to both the host CPU and the offload unit, and including the step of storing connection data for all active connections including those processed at the offload unit and the TCP stack at the host CPU at a connection table in the system memory.

28. (New): The method of claim 26 including the step of determining that an incoming frame is one of the delegated frames; and

determining the frame type of an incoming delegated frame, an incoming frame including an IP packet with a TCP segment being transferred to the TCP stack for processing.

29. (New): The method of claim 26 including the step of determining the validity of an incoming frame and uploading an invalid frame to the TCP stack for legacy processing.

30. (New): The method of claim 27 including the step of processing a valid frame at the offload unit and loading the processed frame directly to the system memory associated with the connection table.

31. (New): The method of claim 30 wherein a valid frame is partially processed at the offload unit when user buffer space associated with the offload unit is available, the valid frame being transferred to the TCP stack for legacy processing when user buffer space is not available.

32. (New): The method of claim 31, wherein a portion of the valid frame processed at the offload unit is limited by a startup limit stored at the delegated connection table.

33. (New): The method of claim 30, wherein each incoming frame includes an acknowledgement (ACK), the offload unit coalescing a plurality of the ACK's before notifying the TCP stack of the status of the delegated connection associated with each received ACK.

34. (New): The method of claim 1, wherein the offload unit compares a sequence number (SN) in each of the incoming frames with a SN stored in the delegated connection table; and

if not equal then the entire frame is uploaded for legacy processing by the TCP stack.